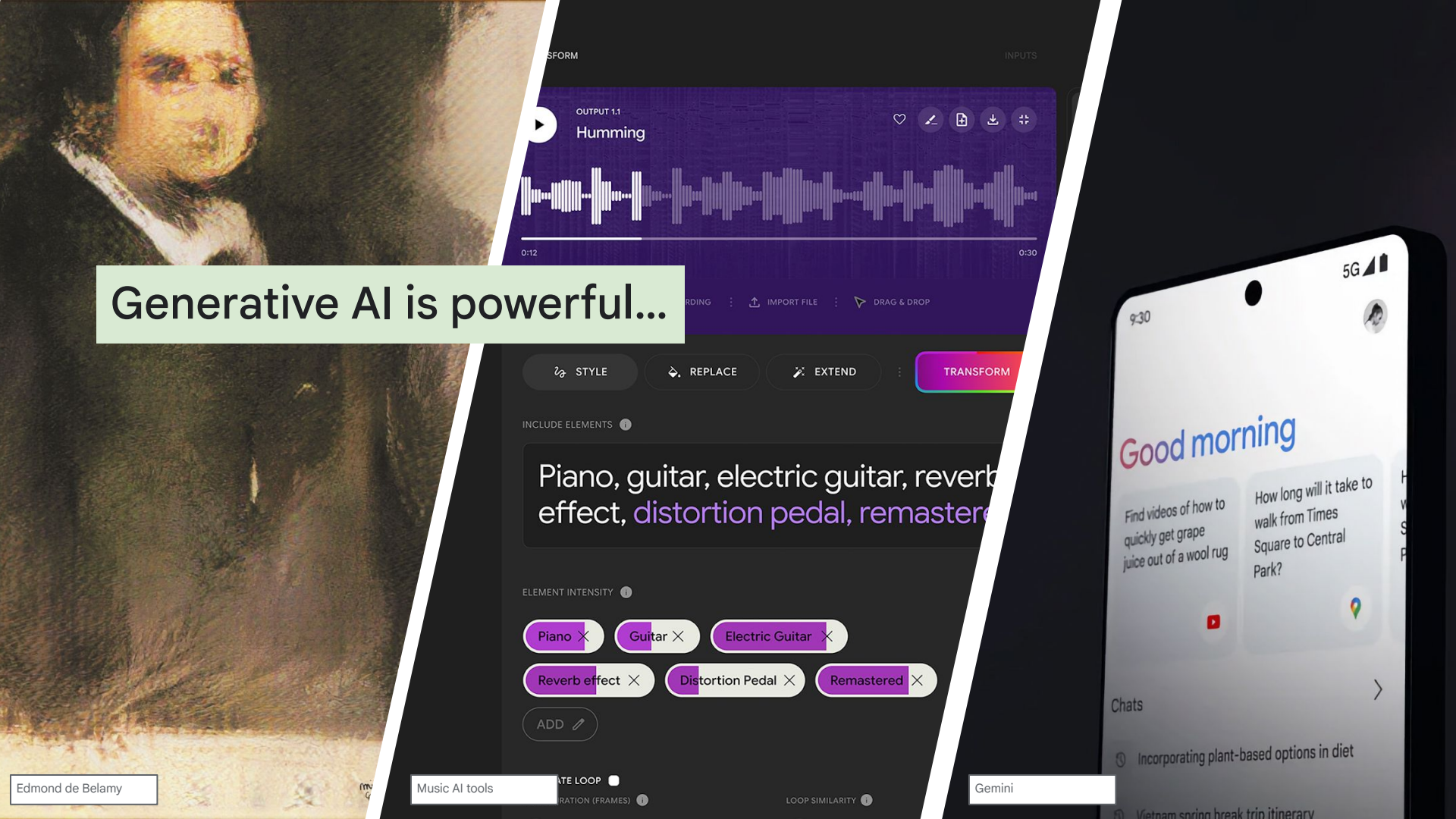
Google DeepMind

SynthID-text: Scalable watermarking for identifying large language model outputs

Abigail See and Sumanth Dathathri
Google DeepMind

01 Motivation


Generative AI is powerful...
Edmond de Belamy
OUTPUT 1.1
Humming
0:12
0:30
IMPORT FILE
DRAG & DROP
STYLE
REPLACE
EXTEND
TRANSFORM
INCLUDE ELEMENTS
Piano, guitar, electric guitar, reverb
effect, distortion pedal, remastere
ELEMENT INTENSITY
Piano
Guitar
Electric Guitar
Reverb effect
Distortion Pedal
Remastered
ADD
Music AI tools
5G
9:30
Good morning
Find videos of how to quickly get grape juice out of a wool rug
How long will it take to walk from Times Square to Central Park?
Chats
Incorporating plant-based options in diet
Gemini

El Malaguero
@ElMalaguero

Pope Francis using a big white puffer jacket in the Vatican City, noon light, screen space Global illumination, lumen reflections, space Reflections, diffraction grading, chromatic aberration, ambient occlusion, realistic photograph, --v 5

10:27 AM · Mar 25, 2023 · **272.1K** Views

On Zoom, 'You're on Mute' Is Now 'Are You Real?'

Scammers used AI to disguise themselves on a video conference and swipe $25 million. Here's how to avoid the same fate.

5 February 2024 at 16:23 GMT

By Parmy Olson
Parmy Olson is a Bloomberg Opinion columnist covering technology. A former reporter for the Wall Street Journal and Forbes, she is author of "We Are Anonymous."

… but can increase the scale of misinformation

Brace Yourself for a Tidal Wave of ChatGPT Email Scams

Thanks to large language models, a single scammer can run hundreds or thousands of cons in parallel, night and day, in every language under the sun.

Identifying AI-generated content is important

- Promote transparency
- Hygiene of the information ecosystem
- Training the next generation of large AI models

Article | Open access | Published: 24 July 2024

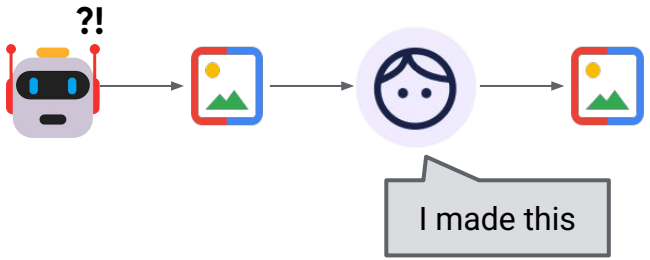
AI models collapse when trained on recursively generated data

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson & Yarin Gal

https://www.nature.com/articles/s41586-024-07566-y

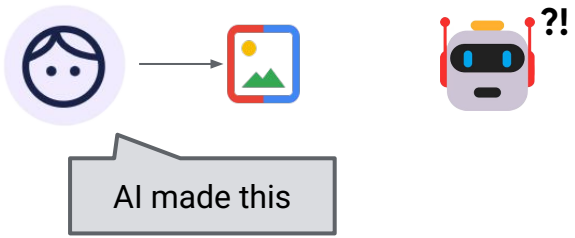
Two goals of an AI-content provenance system

Provenance attestation:
Prevent people falsely claiming ownership of AI-generated content (e.g., prevent using AI content for misinformation)



“scrubbing”

Provenance refutation:
Prevent people falsely claiming that AI has created content (e.g., prevent false claims of harmful AI-generated content)



“spoofing”

Approaches to detecting AI-generated text: **Retrieval-based**

Main idea:

- LLM provider **logs** generated texts
- To determine whether a piece of text was generated by the LLM, **search for it** in the database

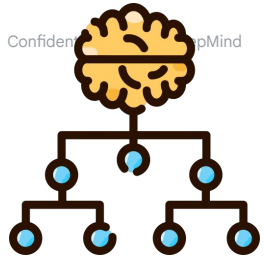
Advantages:

- High precision and recall (for sufficiently unique, unedited texts)

Disadvantages:

- Accuracy tradeoffs when detecting edited texts
- Privacy concerns: cross-user data leakage
- Cannot detect text generated from non-participating (“3rd party”) AI systems

Approaches to detecting AI-generated text: **Classifiers**



Main idea:

- Train a **machine learning classifier** to distinguish AI-generated text from human-written text
- This relies on recognising **different patterns** in AI and human text (e.g. “delve”)

> **Why Does ChatGPT “Delve” So Much? Exploring the Sources of Lexical Overrepresentation in Large Language Models**

Advantages:

- No record-keeping or intervention needed during AI text generation
- Can potentially detect **any** AI-generated content (including by “3rd parties”)

Disadvantages:

- **Inconsistent performance**: out-of-domain text, second-language speakers
- Performance may decrease as LLMs develop; requires frequent retraining
- Can be expensive to run detection (requires running AI classifier model)

Approaches to detecting AI-generated text: **Edit-based watermarking**

Main idea:

- *After the AI text has been generated*, **edit** the text to insert a “mark” that can be detected later
- e.g. synonym substitution or inserting special Unicode characters

Advantages:

- Simple method, computationally inexpensive

Disadvantages:

- Can degrade text quality
- Can leave noticeable artifacts in the text, which might be removable by the user

Approaches to detecting AI-generated text: **Generative watermarking (focus of this talk)**

Main idea:

- *During AI text generation*, **alter the sampling process** so that the generated text has a statistical "mark" that can be detected later

Advantages:

- Can be quality-neutral and imperceptible
- More consistent detection accuracy (e.g. across different types of text)
  - Some accuracy guarantees (e.g. limit probability of false positives)
- Typically simple detection (no model required)

Disadvantages:

- Cannot detect text generated from non-participating ("3rd party") AI systems
- Limited robustness to editing

Objectives of a generative text watermark

Quality-preserving 💎
Watermarking the LLM's output should not degrade its quality.

Detectable 🔎
High statistical certainty when distinguishing watermarked from unwatermarked text.

Robust 💪
Watermark detection should be robust to minor text modifications.

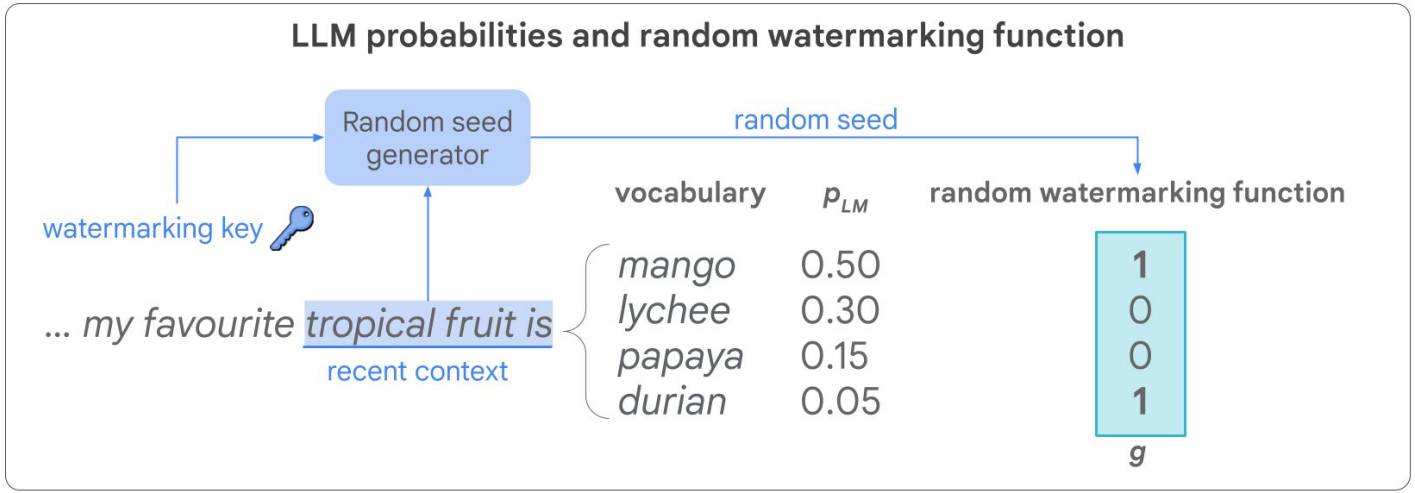
Lightweight 🪶
Watermark generation and detection should have low computational cost and latency. Detection should not require access to LLM.

02 SynthID-text

How LLMs generate text

Main idea of most generative watermarks:
Randomly bias the sampling process towards choosing *some outputs* rather than others. To detect, measure the random bias in the text.

Generating watermarked text with SynthID



💎 **Quality preservation**: In expectation over random seed, $p_{watermarked} = p_{LM}$

Watermark detection

My favourite tropical fruit is mango because of its unique combination of sweetness, tanginess, and juicy texture. The flesh of a ripe mango is like a symphony of flavors, ranging from rich honey and citrus notes to hints of peach and apricot. It's incredibly versatile too, delicious whether eaten fresh, blended into smoothies, or incorporated into both sweet and savory dishes. The aroma alone is enough to transport me to a sun-drenched paradise.

Watermark detection

coinflip value: 1

My favourite tropical fruit is mango because of its unique combination of sweetness, tanginess, and juicy texture. The flesh of a ripe mango is like a symphony of flavors, ranging from rich honey and citrus notes to hints of peach and apricot. It's incredibly versatile too, delicious whether eaten fresh, blended into smoothies, or incorporated into both sweet and savory dishes. The aroma alone is enough to transport me to a sun-drenched paradise.

Watermark detection

coinflip value: 0

My favourite tropical fruit is mango because of its unique combination of sweetness, tanginess, and juicy texture. The flesh of a ripe mango is like a symphony of flavors, ranging from rich honey and citrus notes to hints of peach and apricot. It's incredibly versatile too, delicious whether eaten fresh, blended into smoothies, or incorporated into both sweet and savory dishes. The aroma alone is enough to transport me to a sun-drenched paradise.

Watermark detection

coinflip value: 1

My favourite tropical fruit is mango because of its unique combination of sweetness, tanginess, and juicy texture. The flesh of a ripe mango is like a symphony of flavors, ranging from rich honey and citrus notes to hints of peach and apricot. It's incredibly versatile too, delicious whether eaten fresh, blended into smoothies, or incorporated into both sweet and savory dishes. The aroma alone is enough to transport me to a sun-drenched paradise.

Watermark detection

My favourite tropical fruit is mango because of its unique combination of sweetness, tanginess, and juicy texture. The flesh of a ripe mango is like a symphony of flavors, ranging from rich honey and citrus notes to hints of peach and apricot. It's incredibly versatile too, delicious whether eaten fresh, blended into smoothies, or incorporated into both sweet and savory dishes. The aroma alone is enough to transport me to a sun-drenched paradise.

coinflip values: 1, 0, 1, 1, ..., 1, 0, 1

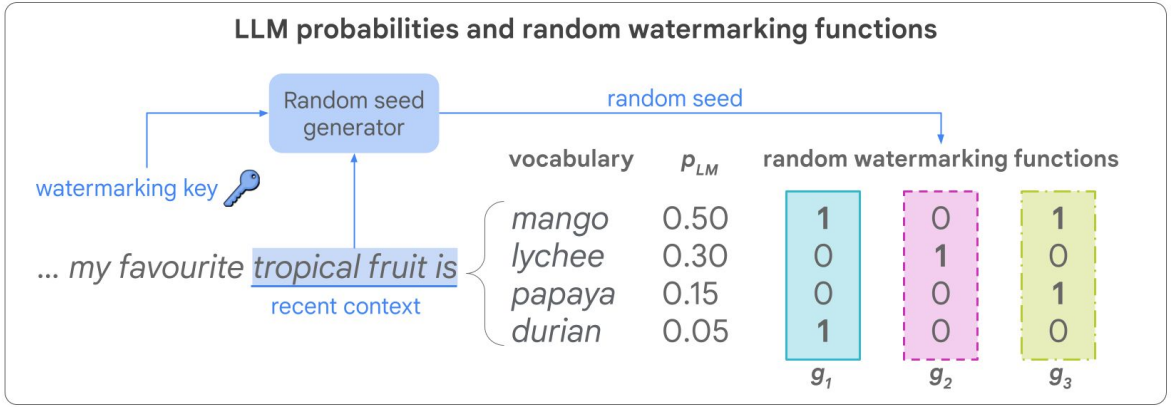
If coinflip values are ~equally many 1s and 0s → no watermark detected

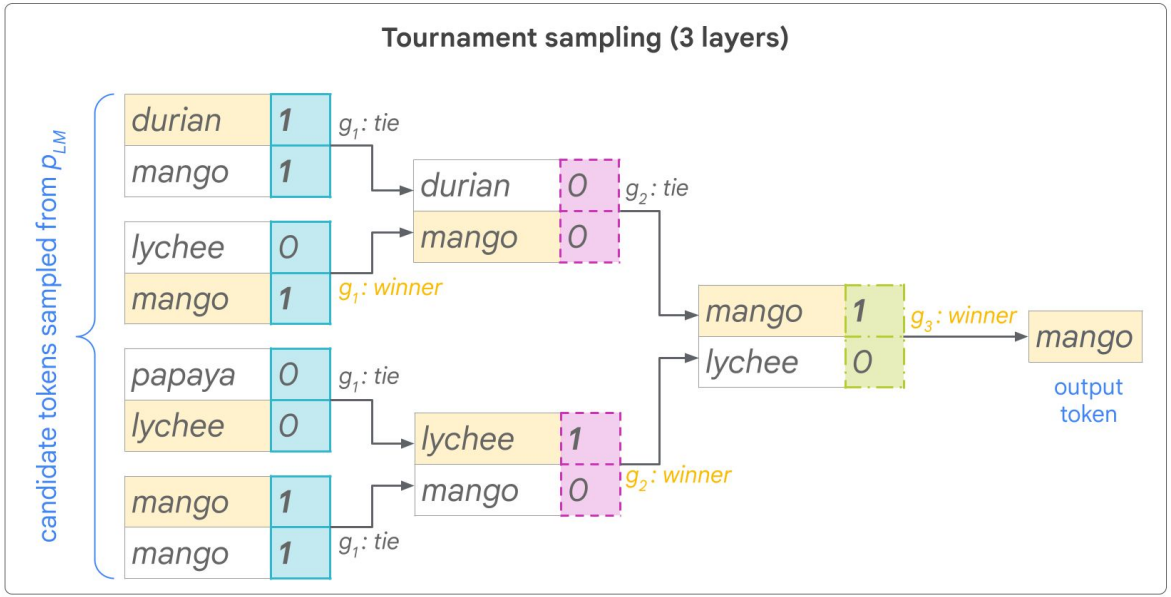
If coinflip values have significantly more 1s than 0s → watermark detected

Two biggest factors for detectability:

1. Text length
2. LLM entropy

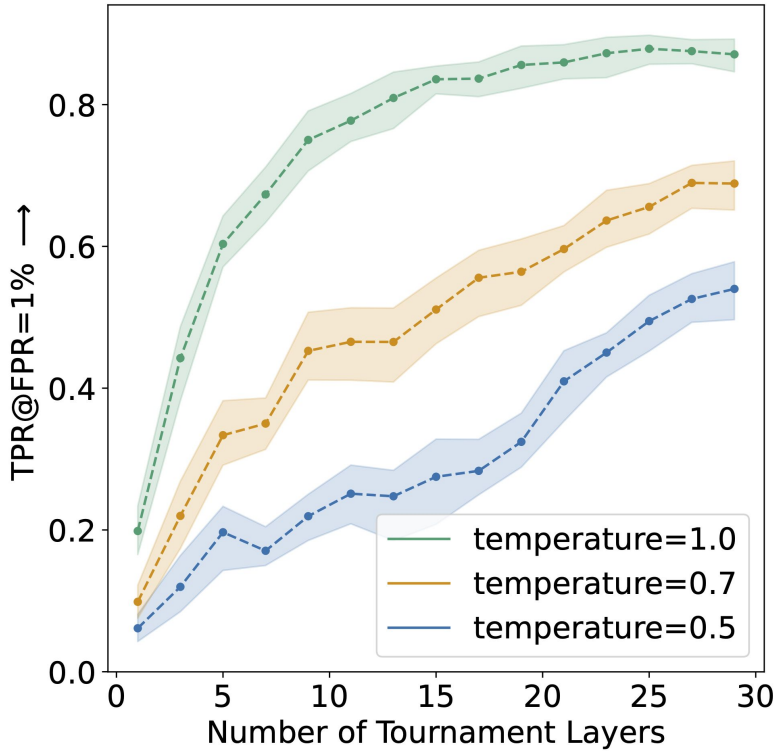
Multi-layer tournament sampling





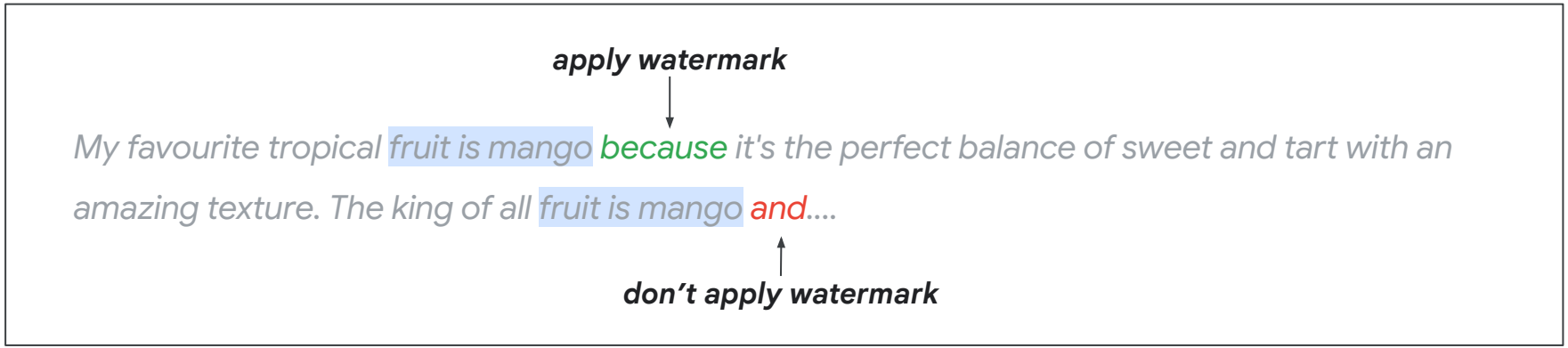
💎 **Quality preservation**:
In expectation over random seed, $p_{watermarked} = p_{LM}$

Detectability increases with more layers



Repeated context masking

Idea: **don't apply watermark** if the preceding **context was previously seen**.
This avoids introducing repeated bias that can affect quality and cause repeating loops.



💎 **Quality preservation**:
Repeated context masking allows us to extend the quality-preserving property from the token level to the *sequence level*.

Similar ideas in:
https://arxiv.org/abs/2310.10669,
https://arxiv.org/abs/2310.07710

Distortionary tournaments

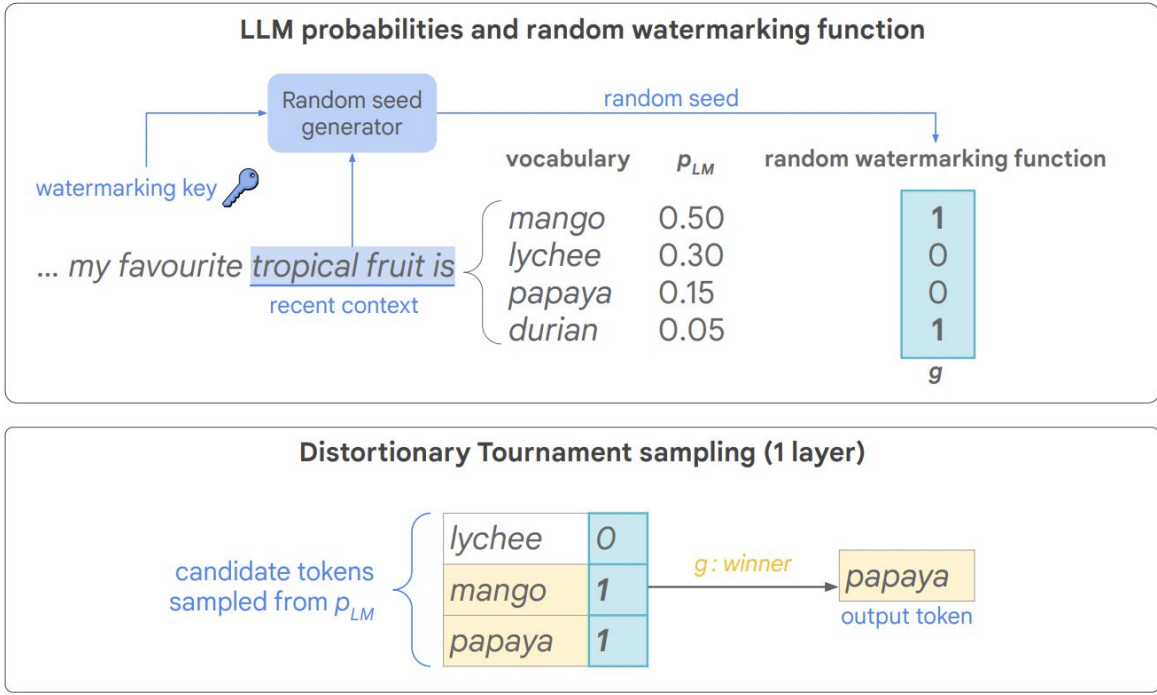
Stronger watermarking, but impact on quality:

In expectation over random choice of $g_1, \ldots, g_m$,

$p_{watermarked} \neq p_{LM}$

Why multi-layer over increasing width of single layer tournament?

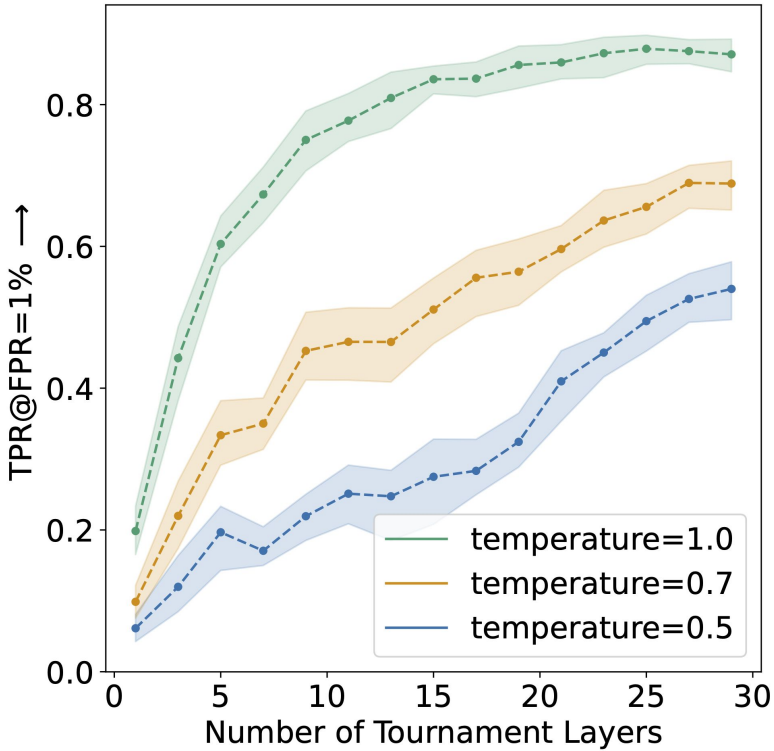
Infinite-width, $p_{watermarked} = Uniform$



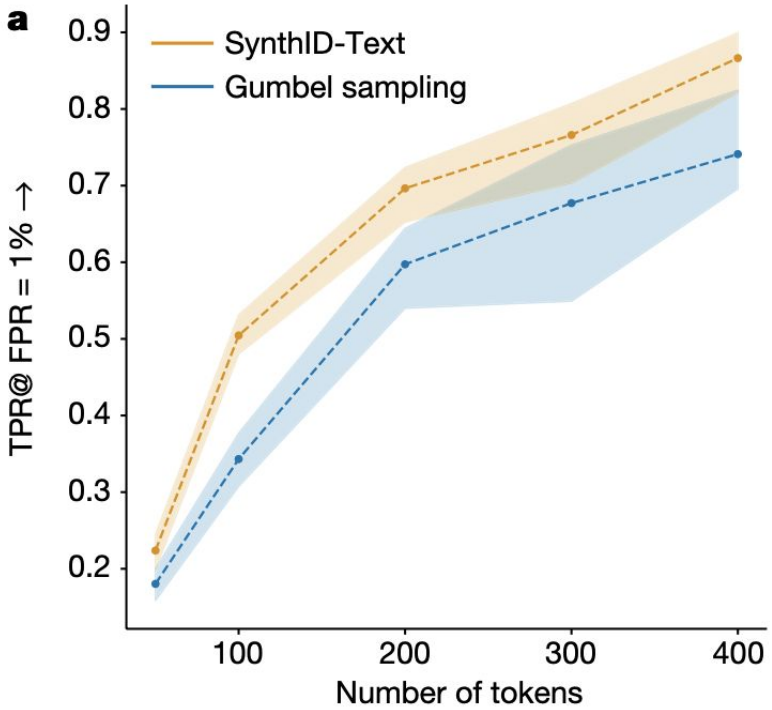
03 Evaluations

More Watermarking Layers, Higher entropy → Better Detectability

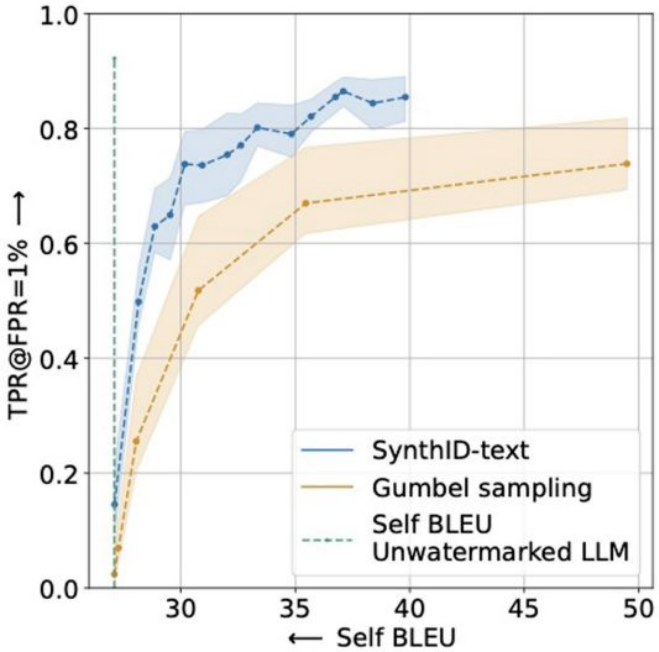
- Detectability is a function of: ***layers*** *x* ***entropy*** *x length of text*



Detectability 🔎



Impact on diversity of generations



→ Watermarking reduces response diversity; smaller reduction for SynthID-Text

Text quality 💎

A/B test with **in-production Gemini web app**:

- **20 million responses**, watermarked/unwatermarked
- User thumbs up / thumbs down rates were **near-identical**

	Thumbs Up Rate Over All Thumbs	Thumbs Down Rate Over All Thumbs
Unwatermarked	0.6824	0.3176
Watermarked	0.6813	0.3187
Difference	-0.16% [-2.16, 1.83] %	0.35% [-3.91, 4.61] %

→ Watermarking leads to no observable difference.

Similar observations for LLM capability benchmarks and detailed human judgments

Latency impact

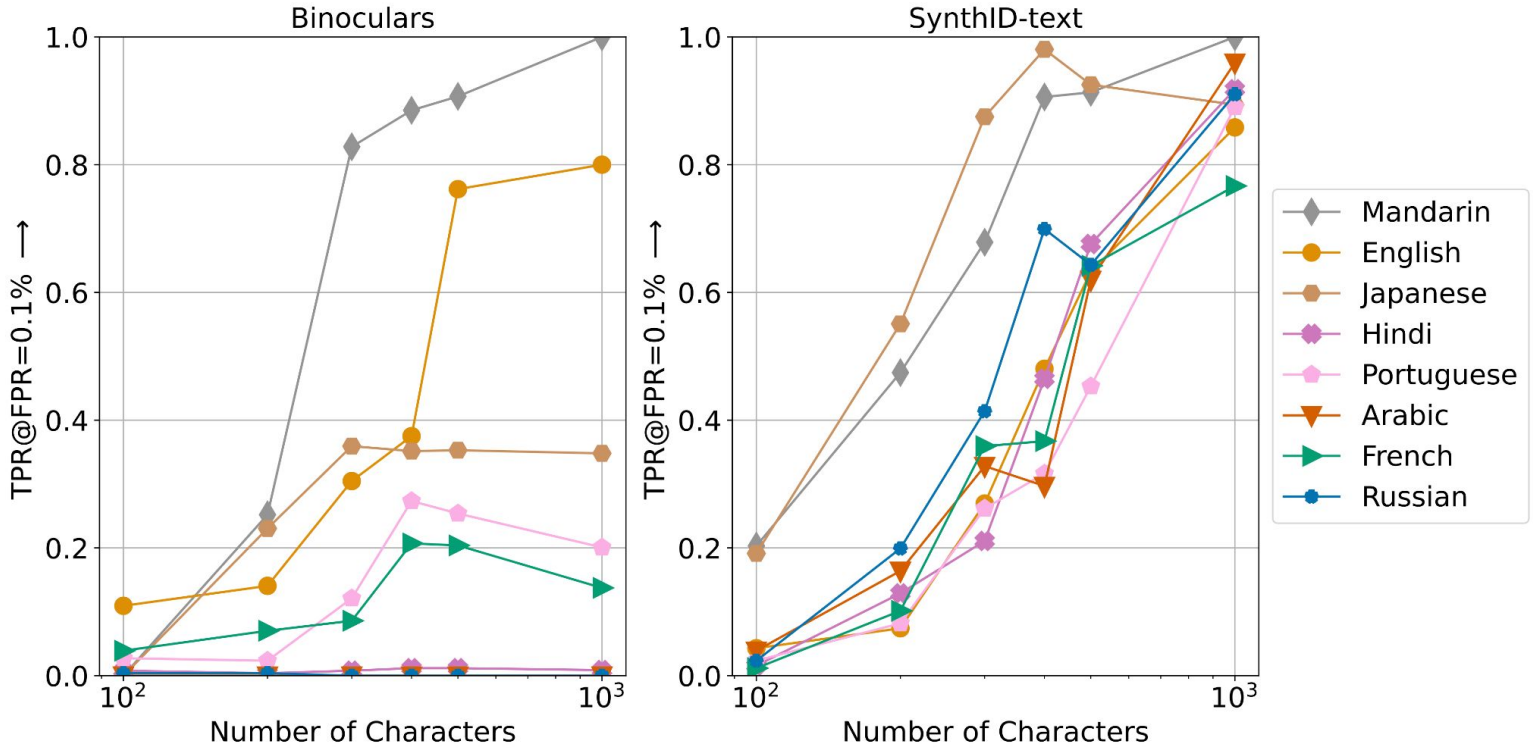
SynthID-text increases generation time by approximately **0.57%** for the Gemma 7B-IT model.

- This is more than for some other generative watermarks (e.g. Gumbel sampling is 0.26%), but still small compared to the overall latency of LLMs.
- Watermarking complexity does not grow with LLM size, so this proportion shrinks for larger LLMs.

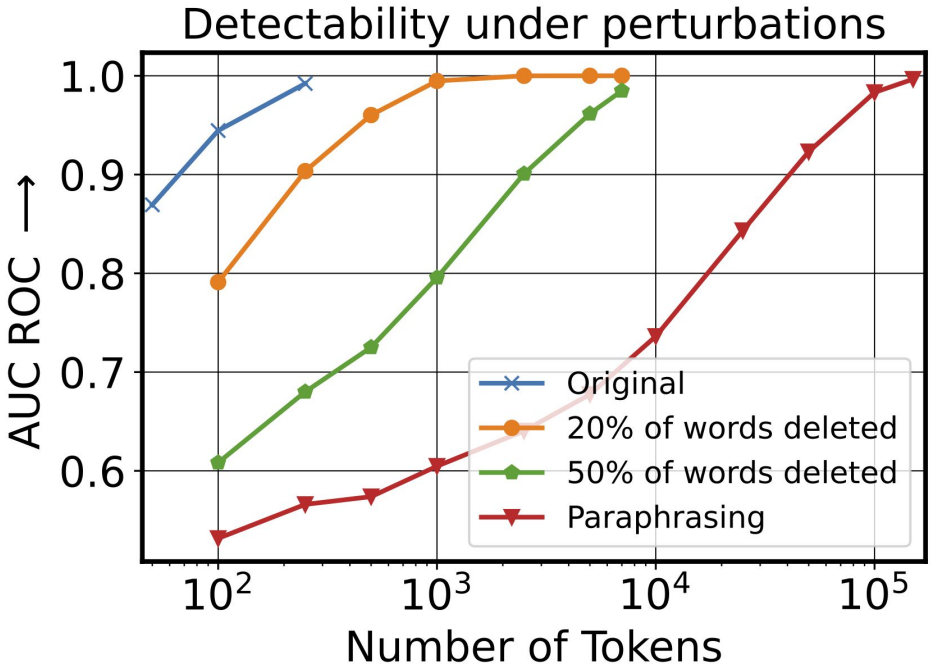
Latency: Compatibility with Speculative Decoding

- Technique for improving the speed of generation from language models
  - (smaller) draft model makes proposals, accept/reject based on target model
- Two versions of quality preserving watermarking with spec. decoding:
  - Preserves watermarking, decreased benefits from speculative sampling
  - Preserves benefits of speculative sampling, weaker watermarking

Detectability comparison with AI text classifiers 🔎



Robustness 💪


Detectability under perturbations
AUC ROC ⟶
1.0
0.9
0.8
0.7
0.6
10^2
10^3
10^4
10^5
Number of Tokens
Original
20% of words deleted
50% of words deleted
Paraphrasing

04 Looking forward

- SynthID-text watermark is applied in Gemini
- SynthID-text watermarking and detection code is open source
  - github.com/google-deepmind/synthid-text
- Wider adoption and coordination towards better hygiene of the ecosystem
- Towards improved accuracy and robustness
  - Watermarking at the level of semantics vs surface level?
- Combining provenance tools together

Thank you.

With contributions from Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, Jamie Hayes, Nidhi Vyas, Majd Al Merey, Jonah Brown-Cohen, Rudy Bunel, Borja Balle, Taylan Cemgil, Zahra Ahmed, Kitty Stacpoole, Ilia Shumailov, Ciprian Baetu, Sven Gowal, Demis Hassabis, Pushmeet Kohli and many others at Google and Google DeepMind!

Google DeepMind